Name: Douglas Takada

WSU ID: 011766707

Summary of how I implemented my proxy:

My implementation is like the tiny server implementation that is given to us. What my proxy server does first is first wait for an input of a port number to listen to with some error checking. Next while it is running, I am looking to accept any connection that a client is trying to connect with the proxy server. When a connection is made, the doit function is run. In here I wait for a request from the connection port of the client. When it is received, some error checking is done and then the request is parsed into the method, uri, and version. Then the uri is further parsed to obtain the host, port, and path of the request. There is some checking done here. If there is not a port specified, then the default port is 80. If a port is specified, then that port is used. But if a port is not specified and the host is either the local host or 127.0.0.1 then the port is labeled the port number that the tiny server is running on. Next the proxy open a connection with the host and port number that the proxy was able to find through parsing the uri. In here the proxy does some error checking to ensure that the when calling read to receive bytes from a socket that has been prematurely closed will not cause a return -1 with error set to ECONNRESET. This code ensures the proxy does not terminate due to this error. Next, the proxy resets the buffer that it has been using to begin writing the request in the proper format to the tiny server. Next the program concatenates all the values in the correct ordering to create a valid get request. In this request the program will also always be sending the headers Connection: close and Proxy-Connection: close as specified in the lab requirements. The proxy also use the correct \r and \n to ensure that the formatting of the request is correct. Once the buffer is filled, it is sent off to the server and waits for the response of the server. When the server responds. The proxy immediately sends all the data that it receives from the server to the client. Once all the data is transferred, the doit function exits and then terminates the connection to the client.

While browsing using Firefox, you may find some of the URL's cannot be accessible using your proxy. Why?

Some of the URL's cannot be accessed with the proxy we have created because our implementation is using the HTTP protocol and most current and up to date URL's use HTTPS protocols. Although these protocols are similar, they are different from each other and cannot send a HTTP request to a website that uses the HTTPS protocol and work.